Common Data Formats For Digital Voting Systems

A Position Paper Submitted by

The Open Source Digital Voting (OSDV) Foundation
TrustTheVote Project

E. John Sebes, Chief Technology Officer

National Institute of Standards & Technology Workshop
October 29-30 2009

1. Introduction

The Open Source Digital Voting (OSDV) Foundation is pleased to present this position paper on common data formats ("CDF"), for the 2009 NIST Workshop on a Common Data Format for Electronic Voting Systems. In this paper, we discuss:

- the needs a common data format should address (*including specific applications*);
- the scope of data covered by a "CDF"; and
- the points within election processes where data should be available in a standard format.

The non-profit OSDV Foundation's flagship effort is the TrustTheVote (TTV) Project. This is a digital public works project intended to re-invent how America votes in a digital democracy. The results – technology everyone can see, touch and try will be freely available, open source, and maintained as a public trust[1].

The TTV Project perspective on CDF has a particular focus on transparency and data interchange among major components of election systems, such voter registration, election management, ballot design, ballot casting, ballot counting, and systems that support audit and timely public reporting. This perspective is from the viewpoint of designers of election technology, and not as manufacturers, distributors, or servicing agents of voting systems.

2. Goals/Requirements of Work on Election Data Formats

Much can be accomplished by initially setting modest goals for requirements concerning CDF, and gaining experience from defining and applying these data formats. We generally advocate agile development in the spirit of the Internet Engineering Task Force[2] (IETF) adage to strive for "rough consensus and running code" as a means to rapid prototyping which often informs further refinement. This cycle is followed by further work towards a more ambitious set of requirements. We advocate that similar philosophy be followed in any effort to generate a CDF, and that the following three guidelines shape the scope and goals of such effort.

2.1 Flexibility and Extensibility

We suggest that initial work on application-specific data formats and early software implementation be done with a goal of flexibility and extensibility of formats. An example is the TrustTheVote Project effort to define a common data format for a voter registration request. This data format is application-specific in the sense that it is required for data exchange between systems that deal with voter registration, such as HAVA-mandated state Digital Voter Registration Systems (DVRS), and third-party or NGO systems that provide online voter registration assistance.

---

1 Unique to this open source effort, the Project's requirements and specifications are driven by a stakeholder community comprised of States' elections directors and officials, and other NGO voting systems domain experts. To date, ~12 States are participating or contributing in some manner to the TrustTheVote Project Technology Core Team's efforts.

2 See generally: **The Tao of IETF: A Novice's Guide to the Internet Engineering Task Force** http://www.ietf.org/tao.html One of the "founding beliefs" (of the IETF) is embodied in an early quote about the IETF from David Clark: "We reject kings, presidents and voting. We believe in rough consensus and running code."

Rather than initially seek a uniform and complete data format, we are working to incrementally develop a "Data Format Definition" (DFD) that largely meets the needs of some states for most scenarios of domestic or overseas voter registration. Our long term plan is to address all States' requirements. We anticipate completing this work will be driven by experience in implementing the DFD with prototypes and trials to evaluate interoperability with existing systems. To wit, we are leveraging the relevant components of the OASIS EML draft version 6.0 §3103 while simultaneously identifying additional data format requirements specific to individual U.S. States.

2.2 Human *and* Machine Readability

We believe that the very important goal of human readability of DFDs (*and datasets expressed using them*) can be met without a trade-off with machine-readability. We believe that readability does not depend on the syntax of a DFD; the literal format of a dataset does not have to be the medium of "readability". Rather, for complex datasets that are inconvenient to read in ASCII, viewing of datasets can be software-assisted using interactive tools ranging from spreadsheet applications used to view CSV or XML, to complex XML reader tools[4], or translators from a richly semantic representation such as XML to a more human readable literal format like YAML[5] Accordingly, human readability can easily be accommodated; so the emphasis of CDF work should be extending existing data format definitions, and developing new ones when required.

2.3 Factor-out Mechanisms for Data Provenance

Provenance of datasets is a very important issue, but one that we believe is orthogonal to the process of iteratively creating and using DFDs to evolve them to a degree of completeness that meets the requirements of actual deployments. At the point where such a DFD is supported by software to be used in an actual deployment, that software should use data provenance methods that are orthogonal to the data representations themselves[6].

3. Scope of Work on Election Data Formats

Continuing our preference for "rough consensus and running code" in the "crawl-phase" of activities[7], we have ideas on scope, including testing, or earlier phase activities in creating and using Data Format Definitions (DFDs) for election software and systems, as follows.

3.1 Broad Scope, Including Voter Registration and Election Management

We believe that any work on CDF should include work on meeting needs relating to voter registration. Digital Voter Registration Systems (DVRS) are a key part of the election IT ecosystem, and have specific requirements for interoperability and data exchange[8]. The same is true for technology used in each of the major steps in the process of preparing for and conducting elections: DVRSs, election management systems (EMSs), ballot design tools, and voting devices for ballot casting and counting. All perform functions of vital interest to members of the public, sometimes in the context of suspicion of malfeasance. As a result, there is potentially significant public benefit from the use of common data formats that enable publishing and transparency, as well as standards-based data interchange.

---

3 See generally: http://bit.ly/HwJiz and more particularly: http://bit.ly/1dr6i6 for the v6.0 Wish List, and specifically: http://bit.ly/j7vf4 for the draft version 6.0 specification including XSD files.

4 There are many such readers; see generally: http://bit.ly/VmtM1

5 See generally: http://www.yaml.org/ and for a fairly good description see: http://en.wikipedia.org/wiki/YAML

6 For instance, employing OASIS standards for digital signatures of XML datasets. See: http://bit.ly/FbJmn

7 In this sense we're referring to the progressive development metaphor of "crawl, walk, run."

8 For example, see the State of California RFP for their planned HAVA compliant digital voter registration system under development by Catalyst Consulting Group of IL (www.catconsult.com), the winning bidder, and the basis for the requirements and specifications for the TrustTheVote Project DVRS. http://bit.ly/y8dZE

3.2 Scope of Interoperability Initially Only for Specific Points in System Architecture

While there is a potentially large scope of election-related data that could have standard data representations, we suggest an initial focus on application-specific use cases related to specific election IT systems. We list several specific cases in Section 4 below, based on the TrustTheVote Project Elections and Voting System Architecture we're developing under advice, counsel and direction from States' Elections Directors, documented on the TrustTheVote Project Wiki[9].

In brief, the main components of this architecture are: a DVRS, an EMS, a Ballot Design Studio, and voting system.[10] Each component has data format requirements in each of 3 basic kinds:

1. interoperability between components of the same kind;
2. interaction between components of different kind, and
3. export of log data and result data that can be used for publication to achieve operational transparency of these systems.

3.3 Scope to Include Log Data to Be Externalized for Broadly Use

Due to our focus on building operational transparency into all elections and voting systems applications, we place equal emphasis on data format definitions for both log data and operational data. We do not characterize event log data as low level and useful only for auditing. For example, a DVRS event log includes information that records transactions that modify a voter record, such as approving or denying a voter registration update request.

Log data about these events is useful for internal accountability, with all administrative users being able to see records of all such transactions, sorted by particular types (*e.g. invalidation of an existing record*). Likewise, transparency can be enabled by redaction and publication of this log data, with considerable public benefit if multiple systems all publish in a common standard data format that enables aggregation and analysis of data published from multiple systems.

3.4 Broad Scope for Publication and Transparency

The scope for publication of information, both log data and operational data, is very broad and certainly not limited publication of election results and related election evidence such as ballot images. The Voter Information Project[11] is an example of public benefit created by a new service that uses existing EMS *precinct definition* data.

3.5 Limited Scope for "Audit-ability" and Interoperability

Our approach to auditing (*which is detailed in an explanatory details appendix available on our TTV Project Wiki*[12]) is to view audit support features not as a requirement for data representation, but as a requirement for software development activities that can re-use, extend, or define data formats as needed for audit support features. With regard to interoperability, we believe that in general, such is a more useful short term goal than conformance. We explain why on our Wiki.

---

9 See: http://bit.ly/hIGfI and for a block diagram see: http://bit.ly/odHiA

10 By an EMS, we mean a data management application for data objects such as precincts, districts, offices, contests, candidates, and elections. By a ballot design studio, we mean software that consumes ballot configuration data and helps humans prepare printable ballot images and e-ballot data assets. A voting system comprises devices for casting and counting ballots, such as accessible ballot marking devices and central and precinct based paper ballot counting devices; tools for managing these device by preparing them with election-specific configuration data, and extracting log data; a tabulator to combine results from multiple ballot counting devices; and an audit-support system for combining log records and other evidence, and enabling review and reporting.

11 See: http://votinginfoproject.org

12 See: the TrustTheVote Project Wiki page(s) on the NIST Workshop on CDF for more supporting content and details: http://bit.ly/2ShFGB

4. What Data to Represent in a "CDF"

Our approach to DFD activities is based on specific use cases of data sharing and/or publication at specific points in the system architecture of our open source project (*being driven by States' elections directors*). The following use cases provide examples of current work, which is documented on our Project Wiki. The list is by no means exhaustive, but indicative of interoperation and interaction between components.

- Voter Registration Request Record – as described previously, we are working on a data format to represent voter registration request, which can be used for data sharing between a state DVRS and an external system that generates the request. OASIS EML draft specification 6.0, §310 provides a basis, while much of our work concerns extensions to support state-specific addenda to FPCA[13].
- Voter Identification Record – used for interoperability between DVRSs to compare voter records in cases of inter-state transfer
- Precinct Address List – an abstract of precincting data that is exported from an EMS, and used by a DVRS to validate and normalize the residence address of a voter and ensure that the address is valid as required for precincting.
- State ID Record – an abstract of a voter identification record, used by a DVRS to cross-check with an external database of state IDs, typically a DMV
- Poll Book Extract – a subset of a voter record (*typically name, address, precinct, party*) exported by a DVRS for use in developing poll-books, ballot mailers, etc., esp. at a *jurisdictional level*;
- Jurisdictional Ballot Configuration Data – produced by an EMS to list one jurisdiction's ballot configurations, for each precinct in the jurisdiction. Used by a ballot design tool as the input to the process of creating printable/paper and/or electronic ballot representation representations for each ballot style corresponding to each ballot configuration;
- Ballot Definition – produced by a ballot design tool to describe ballot representations in terms of ballot configuration items, for ballot casting a counting devices; for example, describing the location on a paper ballot of each voting position, and which content/choice the position represents;
- Balllot Counting Device Recorded-Vote Data – produced by ballot counting device (*precinct or central*), a set of records each representing a set of votes recorded from a single ballot, together with other summary datasets; also used as input by a tabulator device, which combines multiple of these datasets into a single one with summary records that describe contest vote totals.

5. Next Steps

We foresee some portion of next-step activity being performed in existing or upcoming projects to create and/or demonstrate interoperability and/or publication by the extension of existing DFDs (*often in EML or similar, parallel, non-XML syntax*). We recommend that such projects use the tailoring support of EML draft specification v6.0 to experiment with defining the subsets of existing standard DFDs, that can be used to meet specific data representation requirements, and extending them as needed to meet further requirements that are specific to the U.S. or individual States, or to a specific project, initiative, or product.

[13] See: FPCA Registration and Absentee Ballot Request – Federal Postcard Application (FPCA) Standard Form 76A (Rev. 10-2005) http://www.fvap.gov/resources/media/fpca.pdf

For demonstrations of interoperability or data sharing, we expect the most short-term value to be created by technical efforts at the boundaries between large-scale components or devices for voter record management, election management, ballot design, and ballot casting and counting devices.

DFD definition/extension and software demonstration may be the most valuable next steps or interoperability and data sharing purposes, but not for transparency and audit-support purposes. Instead, we believe the most important next steps will be definition of event logging and log data requirements for specific types of components or devices.

We believe the knowledge gap and consensus gap at present are not around structure and format but content. We expect that tangible progress can be made in a small working-group and periodic workshop format. Each group should focus on a particular component or device, defining the transactions and events that should be logged, and for each one, what event specific information should be captured.

For next-step work in both these areas (*interoperability, and enablement of transparency and audit-ability*) we look to NIST coordination and leadership to establish a collaborative public effort, void of bureaucracy wherever possible, lightweight in administration and structure, but conducted in an open transparent manner that encourages public contribution, while quickly reaching consensus. To that end, we believe that leveraging OASIS EML where its current form is of most use, as well as using v6.0 drafts as catalyst where appropriate, can quickly bootstrap this effort. The TrustTheVote Project would be pleased to contribute to such an effort.

6. Summary

The Open Source Digital Voting Foundation's TrustTheVote Project offers seven perspectives on any CDF development effort:

1. Set modest goals for requirements, gain experience from defining and applying these data formats and regularly iterate the process.
2. Conduct all work in the spirit of achieving "*rough consensus and running code*."
3. Strive for flexibility and extensibility, while ensuring human readability of any CDF.
4. Initially factor out mechanisms for data provenance (*revisit them later*).
5. Be holistic in scope. There are a range of system components to address, from registration through audit, in order to produce a CDF of greatest utility.
6. Focus first on interoperability at specific points in the elections ecosystem, with a parallel effort on standardized logging for transparency and audit-ability.
7. Define log data broadly to ensure utility, catalyze transparency, and provide for systems auditing.

Finally, we believe the effort to accomplish all of this is probably best shepherded by NIST, using a organizational structure that is lightweight in administration and operation, designed around agile collaborative working groups, as transparent as possible, while maintaining an objective of achieving consensus and results. And in short, do not allow the great to be the enemy of the good.